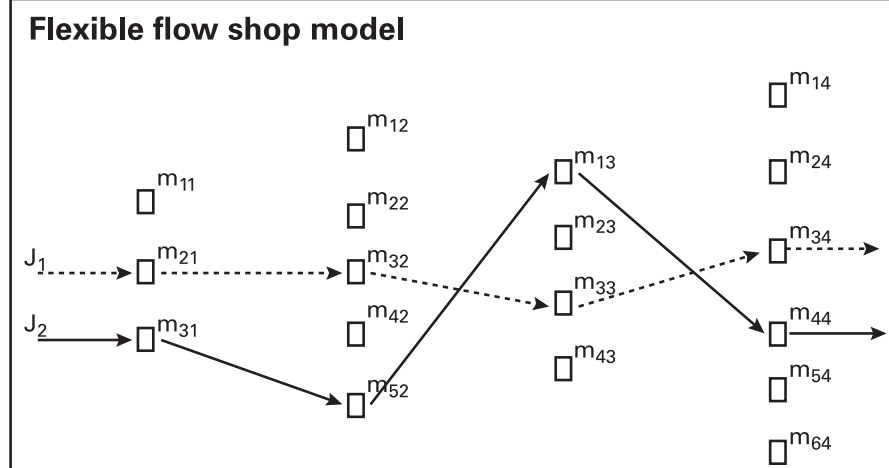


Esempio di sequenziamento di lavorazioni di pezzi (i job J1 e J2) da lavorare su più macchine parallele (cioè funzionalmente equivalenti) suddivise in 4 stadi.



– il setup di un job non dipende da quello lavorato precedentemente le preemption non sono permesse. Il problema HFS, così come la maggior parte dei problemi di scheduling, ha una complessità tale da essere considerato NP-hard (dove NP sta per non deterministic polynomial) poiché è un problema combinatorio, ossia il minimo tempo di completamento di un lotto si trova in corrispondenza di una ben determinata sequenza di job sulle macchine.

Convenzioni, parametri e funzioni obiettivo

La modifica, la rimozione o l'aggiunta delle assunzioni o dei vincoli del problema standard descritto sopra, porta all'ottenimento di diverse varianti del problema HFS. In generale i problemi sono classificati a seconda dei loro shop (α), a seconda delle assunzioni/vincoli (β) e a seconda della funzione obiettivo prescelta (γ). Quindi il problema è descritto dalla terna $\alpha\beta\gamma$. Il parametro α definisce la struttura dello shop, includendo il numero di stages, il numero e le caratteristiche delle macchine per stages. Il parametro β , indica la lista dei vincoli che devono essere considerati nell'affrontare il problema di ottimizzazione. I più comuni sono:

- r_j : indica che il job j , non può essere processato prima della sua release date;
- pmu : indica che tutti i job vengono processati negli stage nello stesso ordine;
- $prec$: indica che ci sono dei vincoli di precedenza tra le operazioni di jobs differenti;
- M_j : indica che il job j può essere processato solo dalle macchine M_j allo stage k ;

Per definire in modo esauriente il problema dello scheduling, devono essere definiti ulteriormente le seguenti quantità:

- Tempi di setup: indica il tempo necessario a riconfigurare la macchina da una lavorazione a un'altra.
- Preemption: se presente, indica la possibilità di interrompere l'esecuzione di un job a favore di un altro a priorità più alta.
- Precedence constraint: indica la presenza di vincoli di precedenza tra i vari job.
- Machine breakdown: indica la presenza di un guasto su una macchina che ne determina il suo inutilizzo.

Inoltre bisogna considerare:

- Tempo di completamento c_{jk} : indica il tempo necessario affinché la lavorazione sul job j sia portata a termine sulla macchina i .
- Lateness L_j : indica la differenza tra il tempo di completamento e il tempo di consegna del job j . Se tale differenza è positiva, si ha un ritardo e se invece è negativa indica un anticipo.
- Tardiness T_j : è definito come $\max(0; L_j)$;
- Earliness E_j : è definito come $\min(0; L_j)$.
- In generale è comune associare al job j un peso w_j che può essere a seconda del caso il costo del job oppure il volume.

Tra le varie funzioni obiettivo da ottimizzare possiamo avere:

- Minimizzazione del makespan definito come il tempo di completamento dell'ultimo job in lavorazione che consente quindi in modo equivalente di massimizzare il coefficiente di utilizzo delle macchine.
- Minimizzazione della Lateness che significa individuare la massima violazione della data di consegna.

- Minimizzazione della Tardiness.
- Minimizzazione del tempo di setup complessivo.

Stato dell'arte

Dato il grande numero di approcci possibili e data la grande varietà del problema si può fare una classificazione in tre grandi classi, cioè quella degli algoritmi esatti quella degli algoritmi euristici e quella degli algoritmi metaeuristici.

Exact algorithm

Senza dubbio la tecnica B&B (branch and bound) è preferita per la risoluzione del problema HFS. Nel caso più semplice, in cui si hanno 2 stage con una sola macchina al primo stage e due al secondo, la tecnica B&B fu proposta per la prima volta da Rao. Successivamente lo stesso problema fu affrontato da Bolat, Harkan e Harby che lo approcciarono con algoritmi genetici euristici e con la tecnica B&B. Il caso opposto fu affrontato da Arthanary, Ramaswamy Mittal e Bagga. Il problema con due stage e qualsiasi numero di macchine identiche al secondo stage fu recentemente studiato e risolto. Inoltre Lee e Kim, proposero un metodo B&B con la minimizzazione della tardiness totale. Invece, il caso in cui il primo stage ha un numero qualsiasi di macchine e il secondo stage ne ha uno solo, è stato studiato da Gupta Hariri e Potts i quali proposero un metodo B&B che è capace di ottenere buone soluzioni in un tempo ragionevole. Successivamente, è stato risolto il problema in cui si hanno due stage senza vincoli sul numero di macchina basato sulla minimizzazione del makespan. In questo caso, è stato implementato un metodo B&B che produce ottime soluzioni per problemi